**Summary of key game functionalities**

**(i) Set up initial game board (including randomised positioning for chit cards):**

**a.** In the UML class diagram, it is illustrated that the Game class creates the actual board used. In the Board class there are three different elements consisting of an array of volcano cards, caves, chit cards, and players. When initially picturing the board, I put only the Volcano cards since that is essentially the board, but to my understanding caves, chit cards and players are also relevant under creating the board. In the Board class there is a createBoard class which takes the volcano cards, chit cards and caves elements into account to join everything together in the desired format to create the board. Finally, the addPlayers method is used to add the number of players in the game to certain caves on the board.

**(ii) Flipping of dragon (chit) cards:**

**a.** Flipping the chit card is a method under the ChitCard class. I've broken the flipping essence into two methods in which the chit card can be flipped up in which it represents the creature and the creature quantity. Under the Chit card class there is also a flipStatus attribute, when a chit card is flipped up the flip status would be set to True, whilst when it is down it is set as False. Flipping a chit card is something the player would do when playing their turn, this is represented through the dependency relationship with the Player class and the ChitCard class.

**(iii) Movement of dragon token based on their current position as well as last flipped chit card:**

**a.** The movement of the dragon token is based on a few different elements with the movePlayer method being the main method. The movePlayer method is used to move the player either backwards, forwards or no movement. The movePlayer method considers the chitMatch method. This method checks whether the creature on the chit card flipped, aligns with the creature of the current position of the player. If there is a match the player moves forward the quantity of the creature which is represented on the chitCard and the currentPosition is incremented accordingly. If there isn't a match the movePlayer method doesn't move the player anywhere. Finally, if the chitCard flipped by the player is a pirate creature the player is moved back regardless of their current position creature with their currentPosition decremented accordingly unless they are in a cave.

**(iv) Change of turn to the next player**

**a.** This functionality is represented through the switchPlayer method under the Player class. Once the current player is having their turn, the switchPlayer method will take place if the chitMatch method returns False. This means the current player taking their turn flipped over a chit card which doesn't correspond to the current volcano card tile creature they are on. This then sets the turnStatus of the current player to False and sets the turnStatus of the next player to True. In the future there may be time components involved which sets a certain amount of time a player has during their turn. When the timer is done the switchPlayer method would again apply in this case.

**(v) Winning the game:**

**a.** It is possible for only a single player to win the game at this point. Winning the game is done when a player goes from their cave around the whole board and

back to their cave. In my Player class I have attributes representing the startPosition, currentPosition and endPosition as integers. Once the player has traversed through all volcano cards with their currentPosition being equal to the endPosition the player has won. This will allow the win method to deliver a message/prompt that a certain player has won, which would then restart the game. In the future there may be features that allow second place, third place etc which would allow only a prompt to pop up without the game having to restart when a single player wins.